

DEEP LEARNING IS COMPETING RANDOM FOREST IN COMPUTATIONAL DOCKING

MOHAMED A. KHAMIS

*Cyber-Physical Systems Lab,
Egypt-Japan University of Science and Technology (E-JUST)
New Borg El-Arab City, Postal Code 21934 Alexandria, Egypt.*

WALID GOMAA

*Cyber-Physical Systems Lab,
Egypt-Japan University of Science and Technology (E-JUST),
New Borg El-Arab City, Postal Code 21934 Alexandria, Egypt.
Faculty of Engineering, Alexandria University, Alexandria 21544, Egypt.*

BASEM GALAL

*Cyber-Physical Systems Lab,
Egypt-Japan University of Science and Technology (E-JUST)
New Borg El-Arab City, Postal Code 21934 Alexandria, Egypt.*

E-mail addresses: mohamed.khamis@ejust.edu.eg, walid.gomaa@ejust.edu.eg,
mllover1992@gmail.com.

ABSTRACT. Computational docking is the core process of computer-aided drug design; it aims at predicting the best orientation and conformation of a small molecule (drug ligand) when bound to a target large receptor molecule (protein) in order to form a stable complex molecule. The docking quality is typically measured by a scoring function: a mathematical predictive model that produces a score representing the binding free energy and hence the stability of the resulting complex molecule.

We analyze the performance of both learning techniques on the scoring power (binding affinity prediction), the ranking power (relative ranking prediction), docking power (identifying the native binding poses among computer-generated decoys), and screening power (classifying true binders versus negative binders) using the PDBbind 2013 database.

For the scoring and ranking powers, the proposed learning scoring functions depend on a wide range of features (energy terms, pharmacophore, intermolecular) that entirely characterize the protein-ligand complexes (about 108 features); these features are extracted from several docking software available in the literature. For the docking and screening powers, the proposed learning scoring functions depend on the intermolecular features of the RF-Score (36 features) to utilize a larger number of training complexes (relative to the large number of decoys in the test set).

For the scoring power, the DL_RF scoring function (arithmetic mean between DL and RF scores) achieves Pearson’s correlation coefficient between the predicted and experimentally measured binding affinities of 0.799 versus 0.758 of the RF scoring function. For the ranking power, the DL scoring function ranks the ligands bound to fixed target protein with accuracy 54% for the high-level ranking (correctly ranking the three ligands bound to the same target protein in a cluster) and with accuracy 78% for the low-level ranking (correctly ranking the best ligand only in the cluster) while the RF scoring function achieves (46% and 62%) respectively. For the docking power, the DL_RF scoring function has a success rate when the three best-scored ligand binding poses are considered within 2 Å root-mean-square-deviation from the native pose of 36.0% versus 30.2% of the RF scoring function. For the screening power, the DL scoring function has an average enrichment factor and success rate at the top 1% level of (2.69 and 6.45%) respectively versus (1.61 and 4.84%) respectively of the RF scoring function.

keywords: Deep learning; Neural networks; Random forest; Drug discovery; Computational docking; Virtual screening.

1. INTRODUCTION

The core process of CADD is *computational docking*. Computational docking is the process of predicting the best orientation and conformation of a small molecule (drug ligand) when bound to a target large receptor molecule (protein) in order to form a stable complex molecule. This amounts to predicting the *binding free energy* (negative value in kcal/mol unit), and hence the stability of the complex molecule resulting from the docking process. A predicted *binding affinity* inhibition constant IC_{50} , K_i , or K_d (positive value in nanomolar unit) is then derived from the predicted binding free energy. This latter value is verified by comparing with the experimentally measured binding affinity. The aim of the docking process is the activation/suppression of the target protein for/from preforming some *functionality*. The *active site* (binding site) of the protein is the *pocket* in which the atoms of the small ligand molecule (key) binds to the nearby amino-acids of the large protein molecule (lock). In molecular docking, a large number of binding poses are evaluated using a *scoring function*. A *scoring function* is a mathematical predictive

model that produces a score that represents the *binding free energy* of a binding pose.

In this paper, we show that a deep learning (DL) scoring function can compete one of the best learning scoring functions that is based on random forest (RF) with respect to the scoring power (binding affinity prediction), the ranking power (relative ranking prediction), docking power (identifying the native binding poses among computer-generated decoys), and screening power (classifying true binders versus negative binders) on the well-known PDBbind benchmark [48] version 2013. Specifically, the contributions presented in this paper are the following: (1) comparing our results with the best classical scoring functions presented in the latest comparative study (CASF-2013) published in mid-2014 [31] whereas most of the similar published work compare with the previous CASF-2007; the methods used for compiling the primary test set have been reformed resulting in an updated test set, i.e., the PDBbind core set (version 2013); this data set is the same size (195 protein-ligand complexes) as the one used in CASF-2007 but its content overlap with only 13%, (2) for the scoring and ranking powers, we use a wide range of features that entirely characterize the protein-ligand complexes; these features include intermolecular features of RF-Score [4] (36 features), energy terms of BALL software [20] (5 features) and energy terms of X-Score [49] (8 features), and pharmacophore features of SLIDE software [53] (59 features), and (3) for the docking and screening powers, we use the intermolecular features of the RF-Score [4] (36 features) to utilize a larger number of training complexes (relative to the large number of decoys in the test set) as well be discussed later.

The remainder of this paper is organized as follows. In Section 2, we give the necessary chemical and molecular biology background with the state-of-the-art of relevant work. Section 3 presents the training and testing data sets, the molecular features, the best classical scoring functions, the proposed learning scoring functions, and the evaluation methods. Section 4 presents the evaluation results of the scoring, ranking, docking, and screening powers of the learning scoring functions. Finally, Section 5 concludes the paper and proposes key points for research extensions.

2. BACKGROUND

2.1. Main capabilities of a scoring function. The most important step in the docking process is scoring the *conformations of a ligand* in the corresponding *binding site* of the receptor protein by using a *scoring function*. The binding affinity prediction using a *scoring function* determines which binding mode is considered the best; a *scoring function* determines which ligand is considered the *most effective drug*.

In [8], the authors present an introduction of the main available molecular docking methods with particular emphasis on the search algorithms and scoring functions. In [18], the authors introduce the protein-ligand docking methods used for structure-based drug design. The authors discuss the fundamental challenges facing these methods and some of the current methodological topics of interest.

There are generally three main capabilities a reliable computational scoring function should satisfy [1]: (1) scoring power: the ability to produce scores for the different binding poses; these scores are supposed to be *linearly correlated* with the experimentally measured binding affinities of the protein-ligand complexes of

known 3D structures, (2) ranking power: the ability to correctly *rank* a given set of ligands with known binding poses when bound to a common protein, and (3) docking power: the ability to identify the *best binding pose* of a given ligand from a set of computationally generated poses when bound to a specific protein. These three performance attributes were referred to by Cheng et al. [11] as scoring power, ranking power, and docking power of a scoring function. In [31], the authors present also the *screening power* which is the ability of a scoring function to identify the true binders to a given target protein among a pool of random molecules.

2.2. Classical versus ML scoring functions. Most scoring functions in use today can be categorized as either [1]: (1) force field-based, (2) empirical-based, or (3) knowledge-based. Force field scores are approximate molecular mechanics interaction energies, consisting of van der Waals and electrostatic components. The parameters that define the intermolecular interactions are derived from experimental data and *ab initio* simulations. Empirical scoring functions adopt a different trend in calculating the binding free energy of the system. The whole energy is assumed to be composed of weighted energy terms. Linear regression methods are used to learn the coefficients of the model. This can be done by fitting the known experimental binding energies to a *training data set*. Finally, a knowledge-based scoring function is based on the theory that large databases of protein-ligand complexes can be statistically mined to deduce rules and models that are implicitly embedded in the data.

As mentioned in [25], the physical-based and knowledge-based scoring functions are *weak predictors* for the binding free energy (and consequently for the binding affinity). Traditional scoring functions assign a common set of weights to the individual energy terms that contribute to the overall energy score. However, as mentioned in [25], the weights assigned to the individual energy terms that contribute to the *overall energy score* are in reality protein-family dependent. Thus, in order to estimate a more accurate binding free energy (and consequently estimate the binding affinity), a scoring function must be trained to derive a unique set of weights for each individual protein-family. Moreover, traditional scoring functions improperly assume that the individual energy terms contribute towards the total binding free energy in an additive manner. Therefore, they predict the binding free energy from a *linear combination* of the individual energy terms. However, this is not theoretically sound [25], since it fails to consider the cooperative effects of the non-covalent interactions. Thus, the scoring function have to model the non-linear relationships among the individual energy terms.

Machine learning is a paradigm shift that has proved itself in the context of virtual screening witnessed by the following. First, improving the prediction ability of the binding affinity than traditional scoring functions (i.e., force field, empirical, etc.). Second, the machine learning approach predicts the binding affinity based on some features of the protein-ligand complex molecule which are naturally available in the literature (e.g., geometric features, physical force field energy terms, pharmacophore features, etc.). In particular, the goal is to learn the relationship between these features and the corresponding experimentally measured binding affinity given some training set of complex molecules. Then, make use of this learned function to predict the binding affinity of new complexes whose features are known but their experimental binding affinity is still unknown. Recently,

there are some non-parametric machine learning techniques used to model the *functional form* of scoring functions given molecular databases, i.e., data-driven (not knowledge-based); each complex structure is represented as a set of features that are relevant in predicting the complex binding affinity.

2.3. Random forest scoring functions. In [4], the authors use random forests [9] to learn how the *atomic-level description* of the complex relates to the experimental binding affinity (RF-Score). Each feature represents the number of occurrences of a particular protein-ligand atom type pair interacting within a certain distance range. Four atom types were selected for proteins (C, N, O, S) and 9 atom types were selected for ligands (C, N, O, S, P, F, Cl, Br, I) considering all the common elemental atom types observed in protein-ligand complexes. The authors in [4] achieve Pearson’s correlation coefficient between the predicted and experimentally measured binding affinities of 0.774 on the PDBbind v2007 core set ($N = 195$ complexes).

In [54], the authors present SFScoreRF that is an RF scoring function for improved affinity prediction of protein-ligand complexes. The authors in [54] achieve Pearson’s correlation coefficient between the predicted and experimentally measured binding affinities of 0.779 on the PDBbind v2007 core set ($N = 195$ complexes).

In [28], the authors present a web platform for large-scale protein-ligand docking namely istar that combined with RF-Score achieves a Pearson’s correlation coefficient between the predicted and experimentally measured binding affinities of 0.855 on the PDBbind v2012 core set ($N = 201$ complexes).

In [5], the authors investigate the impact of the chemical description of the complex on the predictive power of the resulting scoring function. The authors in [5] achieve Pearson’s correlation coefficient between the predicted and experimentally measured binding affinities of 0.803 on the PDBbind v2007 core set ($N = 195$ complexes).

In [29], the authors show that replacing the linear regression used by Cyscore [10] by random forest (RF) can improve prediction performance. In addition, the authors find that given sufficient training samples, RF comprehensively capture the non-linearity between structural features and measured binding affinities. Moreover, the authors prove that incorporating more structural features and training with more samples can both boost RF performance. The authors in [29] use three sets of features: Cyscore [10], AutoDock Vina [45], and RF-Score [4]. Cyscore comprises four numerical features: $\Delta G_{\text{hydrophobic}}$, ΔG_{vdw} , ΔG_{hbond} and $\Delta G_{\text{entropy}}$. AutoDock Vina comprises six numerical features: $Gauss_1$, $Gauss_2$, $Repulsion$, $Hydrophobic$, $HBonding$ and N_{rot} . RF-Score comprises 36 features defined as the occurrence of intermolecular contacts between two elemental atom types. The authors in [29] achieve Pearson’s correlation coefficient between the predicted and experimentally measured binding affinities of 0.803 on the PDBbind v2007 core set ($N = 195$ complexes).

In [51], the authors proposed RF scoring function to predict the protein-ligand binding affinity based on a comprehensive feature set covering protein sequence, binding pocket, ligand structure, and intermolecular interaction. Feature processing and reduction are performed for different protein family datasets. Three family-specific models were constructed for three important protein target families of HIV-1 protease, trypsin and carbonic anhydrase respectively. For comparison,

two generic models including diverse protein families were also built. The evaluation results show that models on family-specific datasets have the superior performance to those on the generic datasets and the Pearson’s correlation coefficient on the test sets are 0.740, 0.874, 0.735 for HIV-1 protease, trypsin and carbonic anhydrase respectively.

In [1], the authors assess the ranking accuracy of ML scoring functions and classical scoring functions using both 2007 and 2010 PDBbind benchmark data sets [48]; working on both diverse and protein-family specific test sets. The best ML scoring function (based on RF) ranks the ligands correctly based on their experimentally measured binding affinities with accuracy 62.5% and identifies the top binding ligand with accuracy 78.1%. For each protein-ligand complex, the authors in [1] extracted features using: X-Score (6 features) [49], AffiScore (30 features) [53, 41, 52], and RF-Score (36 features) [4]. The authors exploit six ML scoring functions: multiple linear regression (MLR), multivariate adaptive regression splines (MARS) [35], k -nearest neighbors (kNN) [40], support vector machines (SVM) [12], random forests (RF) [9], and boosted regression trees (BRT) [39]. The results of the experiments in [1] conclude that utilizing as many relevant features as possible in conjunction with ensemble-based approaches like BRT and RF (which are resilient to over-fitting) is the best option.

In [23], we present a comparative assessment of scoring, ranking, docking, and screening powers of ML scoring functions on the PDBbind v2013 and compared with the classical scoring functions presented in [31]. For a more comprehensive survey about the use of the most prominent ML techniques in computational docking, the readers are referred to [24].

2.4. Beware of RF scoring functions. Despite the superiority of RF scoring functions to predict the experimental binding constants from protein-ligand X-ray structures of the PDBbind dataset, the ranking, docking, and screening powers of RF scoring functions should also be examined. In [17], the authors present RF scoring function trained on simple descriptors that outperforms a prototype scoring function in predicting the binding constants from the atomic coordinates (scoring power test); the authors in [17] achieve Pearson’s correlation coefficient between the predicted and experimentally measured binding affinities of 0.791 on the PDBbind v2007 core set ($N = 195$ complexes). However, the proposed RF scoring function does not discriminate DUD-E [36] actives from decoys in docking experiments (virtual screening power test). Moreover, the proposed RF scoring function is insensitive to docking pose accuracy (docking power test).

In addition, the work presented in [3] proves that ensemble NN scoring functions are more accurate in predicting the binding affinity of protein-ligand complexes than RF. Particularly, neural networks have the ability to approximate any underlying function smoothly in contrast to decision trees that model functions with step changes across decision boundaries. Thus, we propose a deep learning scoring function combined with dropout that can be considered a model averaging technique over a large number of neural networks that outperforms RF scoring function. Hence, the next section presents a comprehensive background on deep learning and its superiority in other application domains.

2.5. Deep Learning. Our work is based on recent successful algorithms in unsupervised feature learning and deep learning [21, 6]. Deep learning is a paradigm that

focuses on learning multiple levels of representation and abstraction that help making sense of data such as images, sound, and text. Deep learning systems achieved state-of-the-art performance on numerous machine learning tasks specially in image and speech recognition.

One of the largest neural networks are built by connecting 16,000 cores [26] to learn multiple levels of representations of images from a large data-set (10 million images selected randomly from the internet). The training was completely unsupervised. The resulting neurons function as detectors for faces, human body, and cat faces. These results proved that computers can learn to detect faces using only unlabeled data.

Recursive neural networks also outperforms the state-of-the-art approaches in segmentation, annotation, and scene classification [42]. The recurrent neural networks are used to build a new system called DeepSpeech [19] that outperformed the previously published results on the widely studied Switchboard Hub5'00 and commercial speech systems.

Deep learning methods are new to the biological field. Recently, deep learning methods are applied to residue-residue contact prediction and disorder prediction [15, 16, 27]. In addition, a couple of deep learning protein structure predictors have been developed, including a multifaceted prediction tool that predicts several protein structural elements in tandem [38].

3. METHODS

3.1. Training and testing data sets. The PDBbind benchmark [48] could be considered the most widely used for binding affinity prediction [29]. Since its first public release in 2004, the PDBbind database has been updated on an annual basis. For instance, PDBbind v2013 provides experimental binding affinity data for 10776 biomolecular complexes in PDB, including 8302 protein-ligand complexes and 2474 other types of complexes. In [34], the authors describe the current methods used for compiling PDBbind. The authors also review some typical applications of PDBbind published in scientific literature. For each protein-ligand complex, the experimentally measured binding affinity either the dissociation constant K_d or inhibition constant K_i was manually collected from its primary literature reference¹.

For detailed information on the compilation of the PDBbind benchmark (version 2013), the reader is referred to [32]. Here we mention briefly the main rules for selecting qualified protein-ligand complexes into the PDBbind refined set (version 2013) [32]: (1) only complexes with crystal structures are accepted, (2) resolution of the complex structure must be better than 2.5 Å, (3) if any fragment on the ligand molecule is missing in the crystal structure, the complex is not included, (4) if any side chain fragment is missing at the protein binding site within 8 Å from the ligand, the complex is not included, (5) complexes with known dissociation constants (K_d) or inhibition constant (K_i) are accepted, while complexes with only half-inhibition or half-effect concentrations (IC_{50} or EC_{50}) values are not included, (6) the binding site on the protein molecule must not contain any non-standard amino acid residues in direct contact with the bound ligand in distance < 5 Å, and

¹The experimentally measured binding affinity either the dissociation constant K_d or inhibition constant K_i used in the present article are extracted from the file named `pdbbind_2013.refined.pdf` that was initially posted on the PDBbind-CN web site (<http://www.pdbbind-cn.org/>).

(7) if the buried surface area of the ligand molecule is below 15% of its total surface area, the complex is not included.

The PDBbind refined set (version 2013) was selected out of 8302 protein-ligand complexes recorded in the PDBbind general set (version 2013) through a complicated process [32] resulting in 2959 protein-ligand complexes. Finally, qualified complexes are clustered by 90% similarity in protein sequences. The binding constants of these complexes $\log K_a$ span nearly 10 orders of magnitude ($\log K_a = 2.07 \sim 11.52$). The resulting 195 complexes are grouped into 65 clusters by protein sequences, namely PDBbind core set (version 2013). Each cluster consists of three complexes, which are referred to as “the best”, “the median”, and “the poorest” by their binding affinities. The binding affinity of the best complex is required to be at least 100 times higher than that of the poorest. Because of the structural diversity of the core set, it is a common practice to use the core set as a test set and the remaining complexes in the refined set as a training set [29].

Decoy ligand binding poses were needed in CASF-2013 [31] to evaluate the docking and screening powers of each scoring function. In CASF-2013, the authors used three popular molecular docking programs, including GOLD (version 5.1, Cambridge Crystallographic Data Center), Surflex implemented in the SYBYL software (version 8.1, CERTARA Inc.) and the molecular docking module implemented in the MOE software (version 2011, Chemical Computing Group). These 3 docking programs use different algorithms for sampling ligand binding poses. The key parameters and settings used in docking are described in detail in [31]; these parameters were chosen to generate diverse rather than converged binding poses.

The outcomes of all three docking programs were combined to obtain a set of ligand binding poses for each complex. Only the binding poses with root-mean-square deviation (RMSD) values lower than 10 Å (in respect to the native binding pose) were considered at subsequent steps. All of these binding poses were grouped by their RMSD values (0 ~ 10 Å) into 10 bins with an interval of 1 Å. The binding poses in each bin were further grouped into up to 10 clusters according to their initial similarities. In each cluster, the binding pose with the lowest internal strain energy was selected as the representative of that cluster. Through the above process, ideally a total of 100 representative decoy ligand poses would be obtained for each complex. However, the number of final selected decoy binding poses was actually lower than 100 for many protein-ligand complexes because of the geometrical constraints of the binding site or the parametric shape of the ligand molecule.

In [23], we used two commands available in the BALL molecular software [20] for decoys pre-processing: (1) *LigandFileSplitter* command that splits the decoys.mol2 file of each complex one molecule per file, and (2) *Converter* command that converts each individual mol2 file to sdf format. We then customized the RF-Score feature extraction code initially available by [4] for taking as an input the protein PDB file and the decoy sdf file and generating the corresponding 36 intermolecular features for every protein-ligand pair.

3.2. Molecular features. For the scoring and ranking powers, the proposed ML scoring functions depend on wide range of features that entirely characterize the protein-ligand complexes. These features include intermolecular features of the RF-Score [4] (36 features), energy terms of the BALL software [20] (5 features) and energy terms of the X-Score [49] (8 features), and pharmacophore features of the SLIDE software [53] (59 features). The RF-Score [4] results in 36 intermolecular

features; the occurrence count of intermolecular contacts between two elemental atom types in every protein-ligand pair. For proteins 4 atom types (C, N, O, S) and for ligands 9 atom types (C, N, O, S, P, F, Cl, Br, I) were selected to generate features considering all the common elemental atom types observed in protein-ligand complexes.

The BALL software [20] results in 5 energy terms; advanced electrostatics, van der Waals, fragmentational solvation, hydrogen bond, and (nRot>14). Two BALL commands are needed to obtain those features; *WaterFinder*² finds the strongly bound waters and *ConstraintsFinder* finds the strongly interacting residues.

The X-Score software [49] results in 8 energy terms;

- (1) ligand molecular weight,
- (2) van der Waals,
- (3) hydrogen bonding,
- (4) hydrophobic pairwise contacts,
- (5) hydrophobic ligand atoms match inside hydrophobic binding site,
- (6) hydrophobic surface area of ligand buried upon binding,
- (7) contribution of the number of rotors for the ligand, and
- (8) octanol-water partition coefficients (log P).

This includes preparing the input PDB file using the *fixpdb* tag of the *xscore* command then preparing the input mol2 file using the *fixmol2* tag, then scoring the protein-ligand complex using the *score* tag, and finally calculate the logp value of the ligand using the *logp* tag.

The SLIDE software [53] results in 59 features of pharmacophore nature;

- (1) ligand hydrophobic cluster points count,
- (2) ligand acceptor cluster points count,
- (3) ligand donor cluster points count,
- (4) ligand doneptor cluster points count,
- (5) protein hydrophobic surface points count,
- (6) protein hydrophobic cluster points count,
- (7) protein acceptor cluster points count,
- (8) protein donor cluster points count,
- (9) protein doneptor cluster points count,
- (10) protein unbumped hydrophobic cluster points count,
- (11) protein metal acceptor points count,
- (12) protein acceptor cluster points remain count,
- (13) protein donor cluster points remain count,
- (14) protein doneptor cluster points remain count,
- (15) protein hydrophobic cluster points remain count,
- (16) protein-ligand hydrophobic contacts,
- (17) protein-ligand H-bond count,
- (18) protein-ligand salt-bridge count,
- (19) number of ligand neighbors,
- (20) number of ligand carbons,
- (21) number of exposed ligand carbons,
- (22) remaining Van der Waals collisions,

²The protein resulting from the WaterFinder tool is used as input for the X-Score and SLIDE software for consistency. Also, since water molecules are often important for binding of ligands, it is advisable to use WaterFinder.

- (23) total Van der Waals overlap (\AA),
- (24) anchor fragment translations,
- (25) side-chain mean-field iterations,
- (26) ligand side-chain rotations,
- (27) protein side-chain rotations,
- (28) number of ligand non-hydrogen atoms,
- (29) total sum of hydrophobicity values for all hydrophobic protein-ligand atom pairs,
- (30) total difference in hydrophobicity values between all hydrophobic protein-ligand atom pairs,
- (31) total difference in hydrophobicity values between all protein-ligand hydrophobic/hydrophilic mismatches,
- (32) total sum of hydrophobicity values for all hydrophilic protein-ligand atom pairs,
- (33) total difference in hydrophobicity values between all hydrophilic protein-ligand atom pairs,
- (34) number of interfacial ligand atoms,
- (35) number of exposed hydro ligand atoms,
- (36) number of ligand flexible bonds,
- (37) number of all interfacial flexible bonds,
- (38) total average hydrophobicity values for all ligand atoms relative to all neighboring hydrophobic protein atoms,
- (39) sum of hydrophobicity values for all hydrophobic protein-ligand atom pairs (in old Slide version),
- (40) number of flexible interfacial ligand bonds,
- (41) sum of the degree of similarity between protein-ligand hydrophobic atom contacts,
- (42) pairwise contact of hydrophilic-hydrophilic protein-ligand contacts,
- (43) pairwise contact of hydrophobic-hydrophilic protein-ligand contacts,
- (44) total of protein hydrophobicity values for protein atoms involved in hydrophobic-hydrophobic contacts,
- (45) sum of hydrophobic atom hydrophobicity values for hydrophobic/hydrophilic mismatch pairs,
- (46) total hydrophilicity of all protein interfacial atoms,
- (47) distance normalized version of protein hydrophobicity values for protein atoms involved in hydrophobic-hydrophobic contacts,
- (48) total of protein hydrophobicity values for protein atoms at the interface,
- (49) distance normalized version of protein hydrophobicity values for protein atoms at the interface,
- (50) total of ligand hydrophilicity values for interfacial ligand atoms,
- (51) number of hydrophobic-hydrophobic contacts,
- (52) number of hydrophobic-hydrophilic contacts,
- (53) increase in hydrophobic environ,
- (54) number of intra target salt bridges,
- (55) number of ligand uncharged polar atoms at the protein-ligand interface which do not have a bonding partner,
- (56) number of protein uncharged polar atoms at the protein-ligand interface which do not have a bonding partner,

- (57) number of ligand charged atoms at the protein-ligand interface which do not have a bonding partner,
- (58) number of protein charged atoms at the protein-ligand interface which do not have a bonding partner, and
- (59) total buried hydrophobic contacts of the ligand upon binding.

This includes dehydrogenating the input pdb file using the *pdbdehydrogen* command then setup the database for biased (ligand-based) template using the *biased* tag of the *setup_dbase* command, then generating the biased (ligand-based) template using the *biased* tag of the *temp_gen* command, then setup the database for unbiased (protein-based) template using the *unbiased* tag of the *slide_setup.pl* command, then generate the unbiased (protein-based) template using the *unbiased* tag of the *temp_gen* command.

A biased template represents known ligand interactions, and an unbiased template represents additional opportunities for making good interactions in the protein binding site. Thus, we merged the two templates into one for screening databases. Firstly, we set up the directory to hold the new merged template using the *merged* tag of the *slide_setup.pl* command. Secondly, we concatenated both of the unbiased and biased templates using the *cat* command. Finally, we run slide based on the merged template using the *merged* tag of the *run_slide* command.

RF-Score was able to retrieve the 36 features on the entire training set 2764 complexes (2959-195) and the entire test set 195 complexes. BALL was able to retrieve the 5 features on 2481 complexes of the training set and 181 complexes of the test set. X-Score was able to retrieve the 8 features on 2507 complexes of the training set and 180 complexes of the test set. SLIDE was able to retrieve the 59 features on 2301 complexes of the training set and 165 complexes of the test set. The final number of common protein-ligand complexes resulting from the four sources of features (RF-Score, BALL, X-Score, and SLIDE) was 2281 complexes in the training set and 164 complexes in the test set.

For the docking and screening powers, the proposed ML scoring functions depend on the intermolecular features of the RF-Score [4] (36 features) for mainly two reasons: (1) utilizing a larger number of training complexes (2764 complexes) relative to the large number of decoys in the test set (docking test set = 15,821 complexes and screening test set = 63,5729 complexes), (2) comparing the presented results with the recently published work in [17] which depends mainly on RF-Score features [4] with several modifications.

3.3. Best classical scoring functions. As mentioned in the introduction section, we compared the performance of the proposed learning scoring functions with the best classical scoring functions presented in the latest comparative study (CASF-2013) published in mid 2014 [31]. The authors in [31] present a panel of 20 scoring functions tested on PDBbind v2013 benchmark. In this panel, 18 scoring functions are implemented in commercial molecular modeling software. In addition, X-Score (version 1.3) is an academic scoring function [49] where the coefficients before each energy term are re-calibrated by [31] on the remaining 2,764 protein-ligand complexes in the PDBbind refined set (version 2013) after removing the 195 protein-ligand complex in the PDBbind core set. In addition, there are three optional modes in X-Score (version 1.3), i.e., $X\text{-Score}^{\text{HM}}$, $X\text{-Score}^{\text{HP}}$, and $X\text{-Score}^{\text{HS}}$, which differ from each other only in the hydrophobic effect term. The performance of $X\text{-Score}^{\text{HM}}$ was slightly better than the other two modes in most tests

in CASF-2013. Thus, X-Score^{HM} was chosen to represent X-Score in CASF-2013. Besides, CASF-2013 introduced a naive scoring function as a reference that uses a single descriptor, i.e., buried solvent-accessible surface area of the ligand molecule upon binding (Δ SAS) which is an estimation of the size of protein-ligand binding interface.

For the scoring power test, X-Score^{HM} was the best classical scoring function where the Pearson’s correlation coefficient between the predicted and experimentally measured binding affinities was 0.614. For the ranking power, X-Score^{HM} ranks the ligands bound to fixed target protein with accuracy 58.5% for the high-level ranking (correctly ranking the three ligands bound to the same target protein in a cluster) and with accuracy 72.3% for the low-level ranking (correctly ranking the best ligand only in the cluster). For fair comparison between the best classical scoring function X-Score^{HM} and our proposed learning scoring functions, we re-generated the previous results on the 164 complexes for which we succeeded to generate features [23]. The following are the updated results of X-Score^{HM}; Pearson’s correlation coefficient = 0.611, high-level ranking = 54%, low-level ranking = 70%.

For the docking power, the best classical scoring function ChemPLP@GOLD has a success rate in identifying the top best-scored ligand binding pose within 2 Å RMSD from the native pose of 81.0%. For the screening power, the best classical scoring function GlideScore-SP has an average enrichment factor and success rate at the top 1% level of 19.54 and 60% respectively.

3.4. Machine learning methods: deep learning & random forest. We analyze the performance of the learning scoring functions with respect to their scoring, ranking, docking, and screening powers. We built a sparse deep auto-encoder to learn multiple layers of representation; the input layer is the 108 features (inter-molecular, energy terms, pharmacophore) that characterize the protein-ligand complexes, then we used greedy layer-wise training algorithm to pre-train our model. In the greedy layer-wise training, each layer is pre-trained to produce a higher level representation of the raw input based on the received representation from the previous layer, which is then fine-tuned using gradient descent.

For the DL scoring function, we used the *Deep learning toolbox* released by Rasmus Berg Palm [37] that implements models for deep learning, e.g., stacked auto-encoders, deep belief network, convolutional neural nets, and various pre-processing functions. We used the dropout technique [43] in order to reduce overfitting. The key idea of this technique is to randomly drop units in training with probability p , then sample the final model from exponential number of thinned networks. At test time, we use single neural network without dropout but each outgoing weights of the units are multiplied by p . This technique improves the performance of our neural network and outperforms other regularization methods.

For the scoring and ranking powers, we trained stacked auto-encoders (SAE) neural network of size $[108 \times 500 \times 500]$ with hyperbolic activation hidden units on unlabeled training data-set of size $[2280 \times 108]$ to learn hierarchical features using greedy layer-wise training method with learning rate = 0.01, and dropout rate = 0.8. Then, we used the trained SAE to initialize neural network of size $[108 \times 500 \times 500 \times 1]$ with linear output layer. Finally, we fine-tuned the neural network using mini-batch gradient descent with batch size = 228, learning rate = 0.002, momentum = 0.9, and dropout rate in fine-tuning = 0.9; this neural network

converges after 670 epochs. For the docking and screening powers, the parameter optimization was done on 4 parameters: learning rate = 0.003, momentum = 0.9, L2 weight decay = 0.005, and dropout rate = 0.9; other parameters are: using 600 epochs, architecture size = $[35 \times 400 \times 400 \times 1]$, and batch size = 691.

For the RF scoring function, we used the *randomForest* function in the *randomForest* package [33] and the RF-Score code available by [4]. In *bagging*, we take a subset of training data (select randomly n training data out of N training data with replacement at each decision node of the tree) and select randomly m input features out of M input features) to train up each tree; after multiple trees are trained, a voting scheme is used to predict testing data; random forest (RF) is one of the most popular *bagging* methods [22]. The tuning parameters of RF include: the n_{tree} that is the number of trees to grow (chosen by 500), and the m_{try} that is the number of variables randomly sampled as candidates at each split; $m_{\text{try}} \in \{2: \text{number of components}\}$. For the scoring and ranking powers: $m_{\text{best}} = 96$. For the docking and screening powers: $m_{\text{best}} = 7$.

For fair comparison between the learning scoring functions, we used the same training set (one complex was excluded at random to make the training set size divisible to equally-sized batches, i.e., the training set size equals DL batch size \times 10). Also, we used the same training complexes shuffling technique for the learning methods in data preparation.

3.5. Evaluation methods of scoring power. As mentioned in CASF-2013, the scoring power refers to the ability of a scoring function to produce binding scores in a linear correlation with the experimentally measured binding affinities. In CASF-2013, this feature was evaluated on the known 3D structures of the 195 protein-ligand complexes in the test set including the original crystal structures and locally optimized complex structures. However, since it is mentioned in CASF-2013 that the performance of most scoring functions retains at basically the same level or gets slightly worse on the optimized complex structures, we found that it is sufficient to compare our results with those of the crystal structures only (for both of the scoring and ranking powers). In CASF-2013, the scoring power of a scoring function was evaluated by the classic Pearson’s correlation coefficient (R) between its binding score and the experimental binding affinities (Eq. 1) and the standard deviation (SD) in regression (Eq. 2) [31]:

$$(1) \quad R = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}}$$

$$(2) \quad SD = \sqrt{\frac{\sum [y_i - (a + b.x_i)]^2}{N - 1}}$$

where, x_i is the binding score computed by a certain scoring function on the i^{th} complex; y_i is the experimental binding affinity of this complex; a and b are the intercept and the slope of the regression line between the computed binding score and the experimental one respectively; the binding constants are all given in the logarithm units ($\log K_a$) [31].

3.6. Evaluation methods of ranking power. As mentioned in CASF-2013, the ranking power refers to the ability of a scoring function to correctly rank the known ligands relative to the same target protein by their binding affinities; the test set used in CASF-2013 consists of 65 clusters of complexes, each of which has 3 complexes formed by the same target protein where the binding affinity of the best complex is required to be at least 100 times higher than that of the poorest.

As mentioned in [1], for test sets that are composed of one protein family (one cluster) bound to a diverse set of ligands, one can interpret the Spearman rank correlation coefficients (R_s) as an alternative measure for ranking accuracy; this measure represents the correlation between ranks based on the predicted binding affinities and the ranks based on the experimentally measured binding affinities; i.e., a scoring function that achieves higher R_s value for some data set is considered more accurate (in terms of ranking power) than its counterparts having smaller R_s values.

As mentioned in CASF-2013, if a scoring function correctly ranked the three complexes in a specific cluster as “the best > the median > the poorest”, one point is recorded for this scoring function, i.e., “high-level” ranking; an overall success rate is computed accordingly over the entire test set [31]. In order to provide an additional index, a “low-level” success rate is considered when a scoring function is able to only rank the best complex as the top one in the cluster regardless of the ranking of the median and the poorest complexes in the same cluster [31].

3.7. Evaluation methods of docking power. As mentioned in CASF-2013, the docking power refers to the ability of a scoring function to identify the native binding pose among computer-generated decoys; ideally, the native binding pose is identified as the one with the best binding score; a set of decoy binding poses (up to 100) was generated for each protein-ligand complex in the PDBbind v2013 core set (195 complexes) by using several molecular docking programs. Each scoring function (in a panel of 20 scoring functions [31]) was applied to score the decoy set of each protein-ligand complex and finds consequently the best-scored binding pose. The decoy set of each complex includes the native binding pose to ensure that there exists at least one correct binding pose. If the RMSD value between the native binding pose and the best-scored binding pose among all decoys plus the native one (computed using Eq. 3) fell below predefined cutoff, e.g., $\text{RMSD} < 2.0 \text{ \AA}$ it is recorded as a successful prediction.

$$(3) \quad \text{RMSD} = \sqrt{\frac{\sum_{i=1}^N [(x_i - x'_i)^2 + (y_i - y'_i)^2 + (z_i - z'_i)^2]}{N}}$$

where (x_i, y_i, z_i) and (x'_i, y'_i, z'_i) are the Cartesian coordinates of the i^{th} atom in two binding poses. Only non-hydrogen atoms in the molecule were considered in [31]. Once this analysis was completed over the entire test set, an overall success rate was computed for every scoring function.

3.8. Evaluation methods of screening power. As mentioned in CASF-2013, the screening power of a scoring function is defined as the ability of a scoring function to identify the true binders to a given target protein among a pool of random molecules (decoys); the screening power was evaluated in a cross-docking trial. The test set includes 65 clusters of PDBbind v2013 core set (195 complexes). Each cluster consists of 3 complexes formed by a certain protein. For each protein,

the 3 known ligands were taken as positives; whereas the other $195 - 3 = 192$ ligands were taken as negatives. For each of the 65 proteins, all 195 ligands were docked into its binding site, resulting in a total of $65 \times 195 = 12,675$ protein-ligand pairs. Since each protein has 3 different structures, the structure of the best complex in each cluster was selected to be the cluster representative. Around 50 representative ligand binding poses were selected for each protein-ligand pair. Each scoring function (in a panel of 20 scoring functions [31]) was applied to score the binding poses of all 195 ligand molecules (including true binders and negatives) of each target protein. For any given ligand, the best-scored binding pose among all available poses was taken as the predicted binding pose and the corresponding binding score was taken as the predicted binding affinity by this scoring function. All 195 ligands were then ranked according to their binding scores in a descending order. The screening power of a scoring function is measured by counting the total number of true binders among the 1%, 5%, and 10% top-ranked ligands. Enrichment factor (EF) is computed using the following equation [31]:

$$(4) \quad EF_{x\%} = \frac{NTB_{x\%}}{NTB_{total} \times x\%}$$

where $NTB_{x\%}$ is the number of true binders observed among the top $x\%$ (where $x = 1, 5$, or 10) candidates selected by a given scoring function. NTB_{total} is the total number of true binders for the given target protein (which is typically 3 for each target protein if there are no cross-binders). As another performance indicator for the screening power in [31]; if the best ligand was found among the 1%, 5%, and 10% top-ranked candidates, a point is counted for the scoring function under test; an overall success rates on the entire test set at the three different levels are computed accordingly.

4. RESULTS AND DISCUSSION

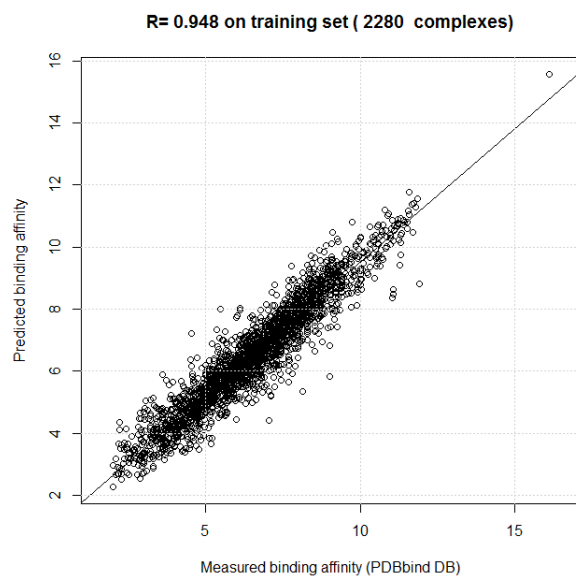
4.1. Evaluation results of scoring power. Fig. 1 presents the correlation coefficient $R = 0.975$ and $R = 0.948$ of the RF and DL scoring functions respectively on the training set (2280 complexes) on the 108 features. Despite the superiority of the RF scoring function on the same training set, the DL scoring function achieves better results on the same test set due to using the dropout technique described earlier which avoids over-fitting.

Fig. 2 presents the correlation coefficient $R = 0.758$ and $R = 0.794$ of the RF and DL scoring functions respectively on the independent test set (164 complexes) on the 108 features. The standard deviation $SD = 1.54$ and $SD = 1.37$ of the RF and DL scoring functions respectively. Despite the RF scoring function outperforms the best classical scoring function X-Score^{HM}, the DL scoring function using the dropout outperforms both scoring functions on the same test set.

The unpaired t test results between the RF-Score and DL-Score could be summarized as follows: (1) the two-tailed P value equals 0.9187, (2) the mean of RF-Score minus DL-Score equals 0.0154 and the 95% confidence interval of this difference lies from -0.2814 to 0.3123, and (3) the intermediate values used in calculations are: $t = 0.1022$, $df = 326$, standard error of difference = 0.151. Despite by conventional criteria this difference is considered to be not statistically significant, the Pearson's correlation coefficient has increased by about 5% of its value (from 0.758 to 0.794) which can be considered an acceptable improvement in this domain. On the other side, the work presented in [47] argues that the statistical significance may not be



(A) RF-Score



(B) DL-Score

FIGURE 1. Correlation coefficient of the RF and DL scoring functions on the training set (2280 complexes) on the 108 features.

very relevant. Basically, the interpretation of the results depends on the context, rather than taking mere numbers. For example, slight increase in performance may

be significant in some applications (for example, life critical), whereas the same increase might be totally irrelevant if the same tool is used in a rather different application.

RF has a built-in tool to measure the importance of individual features across the training set based on the process of “noising up” [4]. For each feature, this consists of randomly permuting its values across out-of-bag (OOB) samples [44] for the current tree and evaluating the mean-square-error (MSE) of these perturbed data ($\text{MSE}_j^{\text{OOB}}$). The higher the increase in error ($\text{MSE}_j^{\text{OOB}} - \text{MSE}^{\text{OOB}}$), the more important the j^{th} feature will be for binding affinity prediction [4]. Fig. 3 presents the relative importance of the first 30 features out of the proposed 108 features. Specifically, this figure shows the increase in error observed when individually noising up each of the 108 features; this is an estimate of the importance of the given feature for binding affinity prediction across the training data [4]. Interestingly, the most 30 important features out of the 108 features span all types of features (energy terms, pharmacophore, intermolecular); this highlights the importance of using features of different nature.

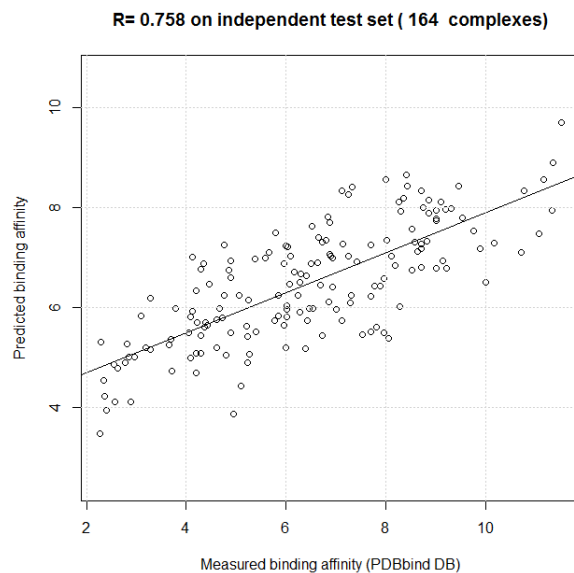
We found that the correlation coefficient $R = 0.999$ of the DL scoring function on the training set (2280 complexes) without using dropout in fine tuning (high over-fitting to the training set), while the correlation coefficient $R = 0.615$ on the independent test set (164 complexes), i.e., much lower than when using dropout, i.e., $R = 0.794$. Thus, Fig. 4 presents the effect of using the dropout technique on the MSE between the predicted and experimentally measured binding affinities versus the training iteration (epoch number) of the DL scoring function. P equals 1 implies no dropout and the optimal performance occurs at P equals 0.9.

We also tried 3 regularization methods (L2, KL-sparsity + L2, and Dropout + L2) and the third one was the best. While the typical value of the dropout rate for hidden units is in the range from 0.5 to 0.8, the best value for our model was 0.9 as shown in Fig. 4. In addition, using dropout requires to increase the size of the network because dropping units in each iteration reduce its capacity. Also, no sparsity penalty is used because dropout already forces the hidden units to be sparse. We tried several architectures with two equally-sized hidden layers, e.g., $[108 \times 400 \times 400 \times 1]$, $[108 \times 500 \times 500 \times 1]$, and $[108 \times 600 \times 600 \times 1]$, and the proposed architecture $[108 \times 500 \times 500 \times 1]$ was a compromise between under-fitting of smaller networks and over-fitting of larger networks.

Fig. 5 presents the correlation coefficient $R = 0.799$ and $R = 0.611$ of the DL_RF (arithmetic mean = predicted scores sum / 2) and X-Score^{HM} scoring functions respectively on the independent test set (164 complexes). The standard deviation $SD = 1.41$ and $SD = 1.78$ of the DL_RF and X-Score^{HM} scoring functions respectively. Fig. 5 highlights that combining more than one learning scoring functions can outperform any individual model. Specifically, these results show the superiority of the learning scoring functions compared to the best classical scoring function X-Score^{HM}.

A test of Pearson’s correlation significance between the predicted and measured binding affinities of the DL_RF can be summarized as following: $t = 16.8879$, $df = 162$, $p\text{-value} < 2.2\text{e-}16$, and the 95% confidence interval lies from 0.7353 to 0.8481.

4.2. Evaluation results of ranking power. In our experiments, the success rates are calculated based on 50 complete clusters (i.e., with 3 complexes of the same protein) after removing any incomplete clusters in the 164 complexes of the test



(A) RF-Score



(B) DL-Score

FIGURE 2. Correlation coefficient of the RF and DL scoring functions on the test set (164 complexes) on the 108 features.

set. The DL scoring function ranks the ligands bound to fixed target protein with accuracy 54% for the high-level ranking (correctly ranking the 3 ligands bound to

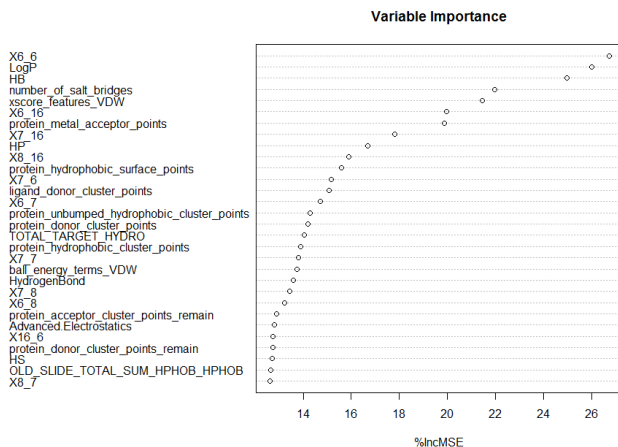


FIGURE 3. Relative importance of the first 30 features out of the 108 features of the RF scoring function.

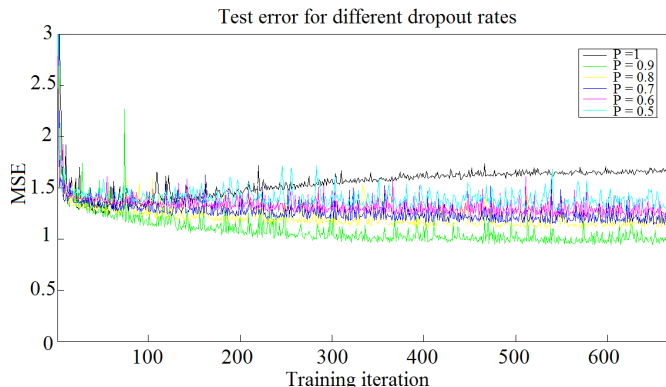


FIGURE 4. Effect of using the dropout technique on the test set mean square error of the DL scoring function.

the same target protein in a cluster) and with accuracy 78% for the low-level ranking (correctly ranking the best ligand only in the cluster) while the RF scoring function achieves (46% and 62%) respectively whereas the best classical scoring function X-Score^{HM} achieves (54% and 70%) respectively. The Spearman's rank correlation coefficients (R_s) between the ranks based on the predicted binding affinities and the ranks based on the experimentally measured binding affinities are as following: DL-RF-Score = 0.789, DL-Score = 0.787, RF-Score = 0.748, and X-Score^{HM} = 0.627.

4.3. Impact of increasing the number of training complexes. In order to perform more validation experiments and in order to study the impact of increasing the number of training complexes on the prediction performance of the learning scoring functions; we used the whole 2445 = (2281 + 164) complexes of PDBbind

v2013; an increasing training set size (489, 978, 1467, 1956) complexes, and an independent test set of 489 complexes. Table 1 presents the performance of the proposed RF-Score with corresponding m_{best} equals (66, 35, 36, 32) respectively versus the DL-RF-Score where the 4 experiments (489, 978, 1467, 1956) training complexes share the following parameters: momentum= 0.9, dropout rate = 0.9, using 600 epochs, architecture size = $[108 \times 500 \times 500 \times 1]$, and batch size = 163. For the training set size equals (489, 978, 1467, 1956) complexes, the DL-Score has the following “learning rate” and “L2 weight decay”: (0.001, 0.01), (0.001, 0.01), (0.001, 0.01), and (0.002, 0.001) respectively.

In Table 1, R is the Pearson’s linear correlation coefficient between the predicted and experimentally measured binding affinities, R_s is the Spearman’s rank correlation coefficient between the ranks based on the predicted binding affinities and the ranks based on the measured binding affinities, RMSE and SD are the root-mean-square-error and standard deviation between the predicted and measured binding affinities respectively. As it is clear from Table 1, the performance improves by increasing the training set size. In addition, the performance of the DL-RF-Score outperforms the performance of the individual RF-Score.

TABLE 1. Impact of increasing the number of training complexes on the performance of the learning scoring functions.

N_{train}	RF-Score				DL-RF-Score			
	R	R_s	RMSE	SD	R	R_s	RMSE	SD
489	0.650	0.644	1.50	1.50	0.658	0.649	1.48	1.48
978	0.685	0.688	1.43	1.43	0.708	0.708	1.39	1.39
1467	0.713	0.713	1.38	1.38	0.722	0.718	1.36	1.36
1956	0.719	0.721	1.37	1.37	0.731	0.729	1.34	1.34

4.4. Evaluation results of docking power. Table 2 presents the performance of the best classical scoring function [31] versus the proposed learning scoring functions in the docking power test when one or more best-scored ligand binding poses are considered. The cutoff of acceptance is that the RMSD value between one best-scored binding pose and the true binding pose is lower than 2.0 Å. The scoring functions are ranked when the top three best-scored ligand poses are considered to match the native pose.

Five complexes were excluded in the docking power test due to the inability of the BALL molecular software for decoy binding poses extraction; these complexes are of PDB IDs: 3ge7, 3gy4, 1o3f, 1o5b, and 1sqa. In addition, the complex of PDB ID = 2d1o is also excluded due to incomplete number of extracted decoy binding poses. Thus, the results are calculated based on the remaining 189 complexes (out of 195 complexes).

As in the scoring and ranking powers test, a DL-based scoring function outperforms the RF scoring function ³. However, the best classical scoring function outperforms the proposed learning scoring functions; these results comply with the PDBbind v2007 results presented in [17] and [2] (when the binding affinity is used

³A shuffling of the training complexes to match the input to the DL scoring function slightly changes the performance of the RF scoring function in the docking and screening power tests; this clarifies the slight change in the results here compared to the ones reported in our recent comparative assessment [23].

for training). Nevertheless, the latter work [2] proves that the ML scoring functions trained to explicitly predict the RMSD values significantly outperform all classical scoring functions. In addition, [50] show that a support vector rank regression (SVRR) algorithm trained with different example datasets, using different training strategies, all achieved increasingly consist accuracies; in contrast, using the same training datasets, traditional support vector classification and regressions algorithms fail to improve comparably the accuracy of conformation prediction; the proposed results suggest that with additional features to indicate the comparative fitness between computed binding conformations, the SVRR algorithm has the potential for more accurate docking scores.

TABLE 2. Success rates in the docking power test when one or more best-scored ligand binding poses are considered.

Scoring function	Success rates (%) on		
	Top pose	Top two poses	Top three poses
ChemPLP@GOLD	81.0	86.7	89.7
DL_RF	15.9	25.4	36.0
RF	18.0	24.9	30.2
DL	13.2	20.6	27.5

Fig. 6 presents the relative importance of the 36 intermolecular features of the RF scoring function in the docking and screening power tests. Among the most important features ($\%incMSE > 20$), we find the occurrence counts of hydrophobic interactions ($x_{6,6}$), of polar-non-polar contacts ($x_{8,6}, x_{7,6}, x_{6,8}, x_{16,6}$), and also of those intermolecular features correlated with hydrogen bonds ($x_{7,8}, x_{8,8}, x_{8,7}, x_{7,7}$) as reported in [4].

4.5. Evaluation results of screening power. Table 3 and Table 4 present the performance of the best classical scoring function [31] versus the proposed learning scoring functions in the screening power test. In Table 3, the scoring functions are ranked by their average enrichment factor obtained at the top 1% level. EF at the top 5% and top 10% levels are considerably lower for all scoring functions because the test set in [31] consists of a rather limited number of true binders (normally three) to each target protein. In Table 4, the scoring functions are ranked by their success rates obtained at the top 1% level. The numbers in brackets are the number of successful cases, for which the upper limit is 65 (for the learning scoring functions the upper limit is 62); 3 incomplete clusters (only 2 true binders per cluster) were excluded in the screening power test due to the inability of the BALL molecular software for decoy binding poses extraction; these complexes are of PDB IDs: 3ge7, 1o3f, and 1sqa. Thus, the results are calculated based on the remaining 62 clusters (out of 65 clusters).

As in the docking power test, a DL-based scoring function outperforms the RF scoring function. However, the best classical scoring function outperforms the proposed learning scoring functions; these results comply with the work presented in [17], where the authors proposed RF scoring function that do not discriminate DUD-E [36] actives from decoys in docking experiments (virtual screening power test). Thus, the current work together with the work presented in [17] highlight the need of checking any novel ML scoring function versus the best classical scoring functions with respect to their screening powers.

Some recent research works proposed ML scoring functions which outperform classical scoring functions in the virtual screening test; this is attributed to the following reason(s): (1) using SVM target-specific models, e.g., [30], [25], and [13]; this can achieve better performance than training using generic database of different protein families, specially virtual screening performance is highly dependent on the target receptor being studied as mentioned in [14], (2) compensating for potential biases (e.g., chemical properties like small-molecule size and polarizability) leads to improvements in the virtual screening performance [14], (3) using DUD-E benchmark [36] which includes a total of 102 target proteins as well as 22,886 known ligands for them; while the adopted test set CASF-2013 [31] includes a rather limited number of true binders (normally three) for each target protein, each target protein in DUD-E has several dozens to several hundreds of known binders which is more suitable for ML scoring functions to train on a large number of known binders for each target protein, and (4) using SVM classifier for discriminating actives from decoys instead of using SVR regressor; e.g., in order to directly compare the performance of the SVM classifier with that of the SVM regressor in [25], the latter was used to rank the DUD InhA data set according to the predicted $\log IC_{50}$ values; the SVM classifier has been shown to be better than the SVM regressor in virtual screening due to the efficient handling of the data imbalance problem; the data imbalance results in a tendency of the model to classify ligands into the negative class, and therefore has high precision (due to small false positives) but low recall (due to large false negatives).

TABLE 3. Enrichment factors in the screening power test.

Scoring function	Enrichment factor		
	Top 1%	Top 5%	Top 10%
GlideScore-SP	19.54	6.27	4.14
DL	2.69	1.40	1.40
RF	1.61	1.40	1.13
DL_RF	1.61	1.29	1.29

TABLE 4. Success rates of finding the best ligand molecule in the screening power test.

Scoring function	Success rates of finding best ligand molecule among (%)		
	Top 1%	Top 5%	Top 10%
GlideScore-SP	60.0 (39)	72.3 (47)	76.9 (50)
DL	6.45 (4)	14.52(9)	25.81(16)
DL_RF	4.84 (3)	14.52(9)	25.81(16)
RF	4.84 (3)	14.52(9)	24.19(15)

4.6. Discussion. In our discussion for the results, we highlight the impact of the dropout technique that averages a large number of thinned neural networks similarly to model averaging of decision trees used in random forests. Thus, dropout is crucial in comparison between the two learning scoring functions. In addition, as mentioned in [3], ensemble NN scoring functions are more accurate in predicting the binding affinity of protein-ligand complexes than RF scoring functions; this is mainly because the ability of neural networks to approximate any underlying function smoothly in contrast to decision trees that model functions with step changes across decision boundaries. This can interpret why the proposed deep learning

scoring function combined with dropout (that is a model averaging technique over a large number of neural networks) outperforms the RF scoring function. On the other side, for learning more robust features using auto-encoders, noise can be added to the input units; in [46], the authors showed that this added noise is helpful in learning deep architectures; dropout also can be interpreted as added noise not only to the input units but also to the hidden units in the neural network and that was very helpful to get more robust features; this was used in the unsupervised feature learning and in fine-tuning to improve the performance of our network.

Many insights have been learned for both the applications and the methodology (deep learning specifically). For instance, the unsupervised pre-training phase played important role in training neural networks especially when we increased the number of hidden units, and during parameter optimization we found that some parameters have larger effect on the performance than others, e.g., dropout rate and learning rate. We also found that using KL-sparsity with dropout is not effective and may lead to worse performance.

For the fact that deep learning performs better than random forest in this particular case and its generalization ability; it is obvious that the efficiency of random forest is mainly due to model averaging of an ensemble of decision trees, though the dropout technique makes averaging more powerful by averaging large number of thinned networks that allowed the DL scoring function to outperform the RF scoring function. In [43], the authors show that dropout improved the performance of neural networks on several supervised learning tasks including computational biology; e.g., testing on an alternative splicing dataset to predict the occurrence of alternative splicing based on RNA features. Thus, in general a deep learning model has many advantages over a random forest model; this is attributed to the deep learning big architecture, unsupervised pre-training, dropout technique, and ability to learn complex functions. Thus, deep learning is expected to outperform random forest in other computational biology prediction tasks.

For the fact that deep learning had worse performance for the training set but better performance for the test set; we have used L2 regularization and dropout to overcome the overfitting on the training dataset; this results in a higher training error and lower testing error compared to the random forest scoring function; this suggests that the random forest model fits the noise on the training data more than the deep learning model while the deep learning model achieves better performance on the test set. For instance, as depicted in Fig. 7 [7], the solid curve gives a perfect fit to the training but it gives poor generalization as a result of fitting noise, on the other hand, the dashed curve gives higher training error but lower test error compared to the solid curve.

5. CONCLUSION

In this paper, we assess the scoring, ranking, docking, and screening powers of deep learning and random forest scoring functions. For the scoring and ranking powers, the proposed learning scoring functions depend on wide range of features (energy terms, pharmacophore, intermolecular) that entirely characterize the protein-ligand complexes. For the docking and screening powers, the proposed learning scoring functions depend on the intermolecular features of the RF-Score.

For the scoring power, the DL.RF scoring function (arithmetic mean between DL and RF scores) achieves Pearson’s correlation coefficient between the predicted

and experimentally measured binding affinities of 0.799 versus 0.758 of the RF scoring function. For the ranking power, the DL scoring function ranks the ligands bound to fixed target protein with accuracy 54% for the high-level ranking and 78% for the low-level ranking while the RF scoring function achieves (46% and 62%) respectively. For the docking power, the DL_RF scoring function has a success rate when the three best-scored ligand binding poses are considered within 2 Å root-mean-square deviation from the native pose of 36.0% versus 30.2% of the RF scoring function. For the screening power, the DL scoring function has an average enrichment factor and success rate at the top 1% level of (2.69 and 6.45%) respectively versus (1.61 and 4.84%) respectively of the RF scoring function.

Machine learning scoring functions in general give the ability to utilize as many relevant features as possible (e.g., geometric features and pharmacophore features) and study the impact of those features on the prediction accuracy. The ensemble-based machine learning approaches (e.g., random forests that are large ensemble of decision trees) are resilient to over-fitting, i.e., give good predictions not only on the training complexes but on any independent test set as well. Similarly, the deep learning machine learning technique using the dropout approach that is an efficient way to average many neural networks is resilient to over-fitting. The classical scoring functions in general fail to model the non-linear relationships among individual energy terms. In addition, the regression coefficients of those individual energy terms are often calibrated based on specific protein family(ies), hence those scoring functions are more prone to over-fitting yielding poor results on independent test sets. Non-parametric machine learning methods (e.g., random forests) have been successful on generic data sets with multiple classes of target proteins.

As presented by [1] and [29] the performance of the ensemble-based machine learning approaches (e.g., random forest, boosted regression trees) is even much improved either by increasing the number of training complexes or increasing the number of features (opposite to the linear regression-based scoring functions). Thus, as a future work we can study the performance of either the ensemble-based machine learning approaches or the deep learning approach using more training complexes, e.g., from the latest release of the PDBbind benchmark (version 2014) and/or using more features extracted from extra molecular softwares.

ACKNOWLEDGMENTS

This work is mainly supported by the Information Technology Industry Development Agency (ITIDA) under the ITAC Program Grant no. CFP#58 and in part by E-JUST Research Fellowship.

REFERENCES

- [1] H. M. Ashtawy and N. R. Mahapatra. A comparative assessment of ranking accuracies of conventional and machine-learning-based scoring functions for protein-ligand binding affinity prediction. *IEEE/ACM Trans. Comput. Biology Bioinformatics*, 9(5):1301–1313, Sept./Oct. 2012.
- [2] H. M. Ashtawy and N. R. Mahapatra. Molecular Docking for Drug Discovery: Machine-Learning Approaches for Native Pose Prediction of Protein-Ligand Complexes. In E. Formenti, R. Tagliaferri, and E. Wit, editors, *Computational Intelligence Methods for Bioinformatics and Biostatistics, 10th International Meeting, CIBB 2013, LNBI 8452*, pages 15–32, Nice, France, jun 2013. Springer International Publishing Switzerland.

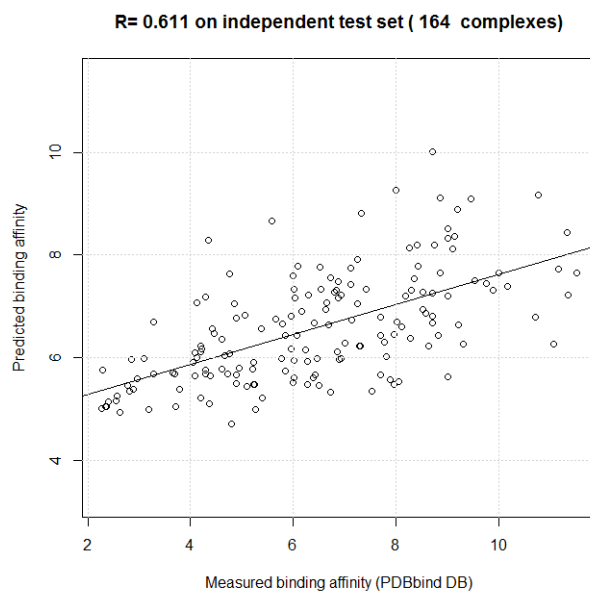
- [3] H. M. Ashtawy and N. R. Mahapatra. BgN-Score and BsN-Score: Bagging and boosting based ensemble neural networks scoring functions for accurate binding affinity prediction of protein-ligand complexes. *BMC Bioinformatics*, 16:(Suppl 4):S8–, 2015.
- [4] P. J. Ballester and J. B. O. Mitchell. A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, Mar. 2010.
- [5] P. J. Ballester, A. Schreyer, and T. L. Blundell. Does a more precise chemical description of protein-ligand complexes lead to more accurate prediction of binding affinity? *Journal of chemical information and modeling*, 54:944–955, 2014.
- [6] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153–, 2007.
- [7] C. M. Bishop. *Neural networks for pattern recognition*. Oxford: Oxford University, 1995.
- [8] N. Brás, N. Cerqueira, S. Sousa, P. Fernandes, and M. Ramos. *Protein Ligand Docking in Drug Discovery*, chapter 11, pages 249–286. Springer International Publishing, Switzerland, 2014.
- [9] L. Breiman. Random forests. *Mach. Learn.*, 45:5–32, 2001.
- [10] Y. Cao and L. Li. Improved protein-ligand binding affinity prediction by using a curvature-dependent surface-area model. *Bioinformatics*, 30(12):1674–1680, 2014.
- [11] T. Cheng, X. Li, Y. Li, Z. Liu, and R. Wang. Comparative assessment of scoring functions on a diverse test set. *J. Chemical Information and Modeling*, 49(4):1079–1093, 2009.
- [12] E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel. *e1071: Miscellaneous Functions of the Department of Statistics (e1071)*, TU Wien, 2010. R Package Version 1.5-24.
- [13] B. Ding, J. Wang, N. Li, and W. Wang. Characterization of small molecule binding. I. Accurate identification of strong inhibitors in virtual screening. *Journal of chemical information and modeling*, 53(1):114–122, 2013.
- [14] J. D. Durrant, A. J. Friedman, K. E. Rogers, and J. A. McCammon. Comparing neural-network scoring functions and the state of the art: applications to common library screening. *Journal of chemical information and modeling*, 53(7):1726–1735, 2013.
- [15] J. Eickholt and J. Cheng. Predicting protein residue-residue contacts using deep networks and boosting. *Bioinformatics*, 28(23):3066–3072, 2012.
- [16] J. Eickholt and J. Cheng. DNDISORDER: predicting protein disorder using boosting and deep networks. *BMC bioinformatics*, 14(1):88–, 2013.
- [17] J. Gabel, J. Desaphy, and D. Rognan. Beware of machine learning-based scoring functions - on the danger of developing black boxes. *Journal of chemical information and modeling*, 54:2807–2815, 2014.
- [18] S. Z. Grinter and X. Zou. Challenges, Applications, and Recent Advances of Protein-Ligand Docking in Structure-Based Drug Design. *Molecules*, 19:10150–10176, 2014.
- [19] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al. Deepspeech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.
- [20] A. Hildebrandt, A. K. Dehof, A. Rurainski, A. Bertsch, M. Schumann, N. C. Toussaint, A. Moll, D. Stöckel, S. Nickels, S. C. Mueller, H.-P. Lenhof, and O. Kohlbacher. BALL - biochemical algorithms library 1.3. *BMC Bioinformatics*, 11:531–, 2010.
- [21] G. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
- [22] R. Ho. *Big Data Machine Learning: Patterns for Predictive Analytics*, 2012.
- [23] M. A. Khamis and W. Gomaa. Comparative Assessment of Machine-Learning Scoring Functions on PDBbind 2013. *Engineering Applications of Artificial Intelligence*, pages 136–151, 2015.
- [24] M. A. Khamis, W. Gomaa, and W. F. Ahmed. Machine learning in computational docking. *Artificial Intelligence In Medicine*, 63:135–152, 2015.
- [25] S. L. Kinnings, N. Liu, P. J. Tonge, R. M. Jackson, L. Xie, and P. E. Bourne. A machine learning based method to improve docking scoring functions and its application to drug repurposing. *J. Chem. Inf. Model*, 51(2):408–419, Feb. 2011.
- [26] Q. V. Le. Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE, 2013.

- [27] P. D. Lena, K. Nagata, and P. Baldi. Deep architectures for protein contact map prediction. *Bioinformatics*, 28(19):2449–2457, 2012.
- [28] H. Li, K.-S. Leung, P. J. Ballester, and M.-H. Wong. istar: A Web Platform for Large-Scale Protein-Ligand Docking. *PLoS ONE*, 9(1):e85678–, 2014.
- [29] H. Li, K.-S. Leung, M.-H. Wong, and P. J. Ballester. Substituting random forest for multiple linear regression improves binding affinity prediction of scoring functions: Cyscore as a case study. *BMC Bioinformatics*, 15:291–, 2014.
- [30] L. Li, B. Wang, and S. O. Meroueh. Support vector regression scoring of receptor-ligand complexes for rank-ordering and virtual screening of chemical libraries. *Journal of Chemical Information and Modeling*, 51(9):2132–2138, 2011.
- [31] Y. Li, L. Han, Z. Liu, and R. Wang. Comparative assessment of scoring functions on an updated benchmark: 2. evaluation methods and general results. *J. Chemical Information and Modeling*, 54:1717–1736, 2014.
- [32] Y. Li, Z. Liu, J. Li, L. Han, J. Liu, Z. Zhao, and R. Wang. Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set. *J. Chemical Information and Modeling*, 54(6):1700–1716, 2014.
- [33] A. Liaw and M. Wiener. Classification and Regression by randomForest. *R News*, 2(3):18–22, 2002.
- [34] Z. Liu et al. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics*, 31(3):405–412, 2015.
- [35] S. Milborrow. *Earth: Multivariate Adaptive Regression Spline Models*. Derived from Mda:mars by Trevor Hastie and R. Tibshirani, 2010. R Package Version 2.4-5.
- [36] M. M. Mysinger, M. Carchia, J. J. Irwin, and B. K. Shoichet. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *Journal of Medicinal Chemistry*, 55:6582–6594, 2012.
- [37] R. B. Palm. Prediction as a candidate for learning deep hierarchical models of data. Master’s thesis, Technical University of Denmark, Kongens Lyngby, 2012.
- [38] Y. Qi, M. Oja, J. Weston, and W. S. Noble. A unified multitask architecture for predicting local protein properties. *PloS one*, 7(3):e32235–, 2012.
- [39] G. Ridgeway. *GBM: Generalized Boosted Regression Models*, 2010. R Package Version 1.6-3.1.
- [40] K. Schliep and K. Hechenbichler. *kknn: Weighted k-Nearest Neighbors*, 2010. R Package Version 1.0-8.
- [41] V. Schnecke and L. Kuhn. Virtual Screening with Solvation and Ligand-Induced Complementarity. *Virtual Screening: An Alternative or Complement to High Throughput Screening?*, pages 171–190, 2002.
- [42] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*, pages 129–136, 2011.
- [43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [44] V. Svetnik et al. Random forest: a classification and regression tool for compound classification and QSAR modeling. *Journal of chemical information and computer sciences*, 43(6):1947–1958, 2003.
- [45] O. Trott and A. J. Olson. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010.
- [46] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. of 25th International Conference on Machine Learning (ICML 2008)*, pages 1096–1103. ACM, 2008.
- [47] K. L. Wagstaff. Machine learning that matters. In *Proc. of 29th International Conference on Machine Learning (ICML 2012)*, Edinburgh, Scotland, UK, 2012.
- [48] R. Wang, X. Fang, Y. Lu, and S. Wang. The PDBbind Database: Collection of Binding Affinities for Protein-Ligand Complexes with Known Three-Dimensional Structures. *J. Medicinal Chemistry*, 47(12):2977–2980, 2004.
- [49] R. Wang, L. Lai, and S. Wang. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *J. Computer-Aided Molecular Design*, 16:11–26, 2002.

- [50] W. Wang, W. He, X. Zhou, and X. Chen. Optimization of molecular docking scores with support vector rank regression. *Proteins: Structure, Function, and Bioinformatics*, 81(8):1386–1398, 2013.
 - [51] Y. Wang, Y. Guo, Q. Kuang, X. Pu, Y. Ji, Z. Zhang, and M. Li. A comparative study of family-specific protein-ligand complex affinity prediction based on random forest approach. *Journal of Computer-Aided Molecular Design*, pages 1–12, 2014.
 - [52] M. Zavodszky and L. Kuhn. Side-chain flexibility in protein-ligand binding: The minimal rotation hypothesis. *Protein Science*, 14(4):1104–1114, 2005.
 - [53] M. Zavodszky, P. Sanschagrin, L. Kuhn, and R. Korde. Distilling the essential features of a protein surface for improving protein-ligand docking, scoring, and virtual screening. *J. Computer-Aided Molecular Design*, 16:883–902, 2002.
 - [54] D. Zilian and C. Sotriffer. SFCscoreRF: a random forest-based scoring function for improved affinity prediction of protein-ligand complexes. *Journal of chemical information and modeling*, 53(8):1923–1933, 2013.
-



(A) DL_RF-Score



(B) X-Score^{HM}

FIGURE 5. Correlation coefficient of the DL_RF and X-Score^{HM} scoring functions on the test set (164 complexes) on the 108 features.

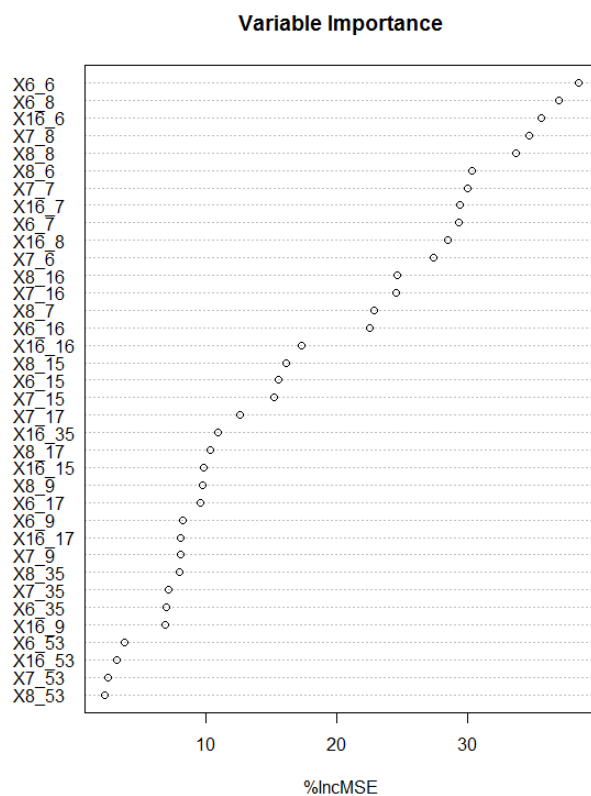


FIGURE 6. Relative importance of the 36 intermolecular features of the RF scoring function in the docking and screening power tests.

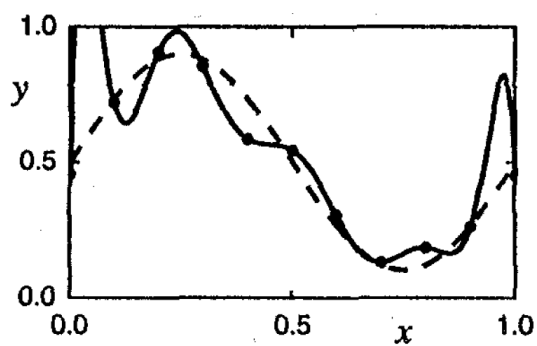


FIGURE 7. An overfitting example.